DETAILED ACTION

Remarks

1. This Office action is responsive to the Request for Continued Examination (RCE) filed under 37 CFR §1.53(d) for the instant application on 3/28/08. Applicants have properly set forth the RCE, which has been entered into the application, and an examination on the merits follows herewith.

Claims 1, 4, 6-15, 17-20, 23-26, 29-38, 40, and 41 have been examined and rejected. This Office action is responsive to the amendment filed on 3/28/08, which has been entered in the above identified application.

Claim Objections

2. Claims 33-38, 40, and 41 are objected to because of the following informalities:

a. On *[line 5]* of claim 33, Examiner suggests changing "the plurality of applications" to --a plurality of applications--.

Appropriate correction is required.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the

applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 1, 6-8, 10-14, 17-20, 24-26, 29, 31-34, 36-38 and 41 are rejected under 35 U.S.C. 102(e) as being anticipated by Lin et al. (U.S. Patent No. 6,369,835 B1).

Claims 1, 6-8, 10, 11 (Method)

Claims 12-14, 17-19 (Method)

Claims 20, 24, 25 (Computer Readable Media)

4-1. Regarding 1, 12, 20, Lin discloses a method and computer-readable media comprising examining a plurality of nodes within a media timeline, by disclosing transforming slides for a slide show presentation into movie data and saving the movie data in a movie file for display as a movie by any program capable of playing the movie file *[column 1, lines 6-12]*. Movie data comprises slide video samples, transition video samples, effect samples, and multiple types of media such as video, sound, and sprite in which references to them are disposed on a video track contained in a video media *[column 3, lines 11-37]*.

Lin teaches wherein the media timeline is for exposure over an application programming interface (API), by disclosing that a movie application programming interface may be selected to save the movie data in the movie file *[column 3, lines 3-10]*.

Lin teaches one or more said nodes reference respective media and dividing the media timeline into one or more presentations, wherein each said presentation describes a rendering of the media for a particular interval of time, by disclosing that the video track references a segment on the track occupied by a sample to a corresponding order and timing of the corresponding objects in the slide show presentation. Each track is a timeline with marks (references) that denote when the movie API should begin playing samples from an identified media container *[column 10, lines 43-54]*. Thus, each media container represents a presentation of a particular media.

Lin teaches wherein each said presentation describes a collection of software components that, when executed, provides the described rendering of the media for the particular interval of time, by disclosing that each track is a timeline with marks (references) that denote when the movie API should begin playing samples from an identified media container *[column 10, lines 43-54]*. Software components include transitions and effects *[column 14, lines 44-55]*, audio data such as background audio *[column 12, lines 53-65]*, voice narration *[column 13, lines 3-17]*, and sound effects *[column 13, lines 18-37]*, and video clips from other applications *[column 18, lines 20-24]*.

Lin teaches wherein the collection of software components include a transform by disclosing that the movie API provides editing the movie data such as producing/calling up a video effect including fade, wipe, move, swivel, and spiral as well as selecting the number of colors, size, and position *[column 3, lines 38-50]*. Lin teaches wherein the collection of software components comprise at least one of a timeline

source, a media source, a media session, a media engine, a source resolver, and a media sink, by disclosing using a track data structure that references the segment on the track occupied by a sample to the corresponding order and timing of the corresponding object in the presentation. Each track is a timeline with marks that denote when the movie API should begin playing samples from an identified media *[column 10, lines 43-54]*. Additionally, software components may include transitions and effects *[column 14, lines 44-55]*, audio data such as background audio *[column 12, lines 53-65]*, voice narration *[column 13, lines 3-17]*, and sound effects *[column 13, lines 18-37]*, and video clips from other applications *[column 18, lines 20-24]*.

Lin teaches further comprising loading each said software component described by a first said collection, executing each said software component described by the first said collection, and loading each said software component described by a second said collection, by disclosing that each track is a timeline with marks (references) that denote when the movie API should begin playing samples from an identified media container *[column 10, lines 43-54]*. Thus, each of the software components would be loaded and executed based on a particular time interval.

As per claim 12, Lin teaches receiving a call from an application over the API for rendering the media timeline *[column 10, lines 48-54]*.

4-2. Regarding claim 6, Lin discloses method as described in claim 1, wherein each said software component that is described by the second said collection is loaded during the executing of the first said collection *[column 8, lines 43-48: audio]*.

4-3. Regarding claim 7, Lin discloses method as described in claim 1, further comprising receiving a request from the application over the API to render the media timeline *[column 10, lines 48-54]*.

4-4. Regarding claims 8, 17, and 24, Lin discloses method and source as described in claims 1, 12, and 20, wherein at least one said node is configured to reference an effect to be applied to an output of said media referenced by the node *[column 10, lines 55-65]*.

4-5. Regarding claims 10 and 25, Lin discloses method and source as described in claims 1 and 20, wherein at least one said node is configured for communication of events to another said node such that a change may be made to the media timeline while the media timeline is rendered *[column 19, lines 46-52]*.

4-6. Regarding claims 11 and 19, Lin discloses one or more computer readable media comprising computer executable instruction that, when executed on a computer, direct the computer to perform the method of claims 1 and 12 *[column 5, lines 48-54]*.

4-7. Regarding claim 13, Lin discloses method as described in claim 12, wherein the rendering further comprises examining the media timeline *[column 19, lines 46-49]*.

4-8. Regarding claim 14, Lin discloses method as described in claim 12, wherein each said collection does not change for the particular interval of time described by a respective said presentation *[column 8, lines 11-20]*.

4-9. Regarding claim 18, Lin discloses method as described in claim 12, wherein at least one said node is configured for communication of events to another said node such that a change may be made to the media timeline while the media timeline is rendered *[column 19, lines 46-52]*.

Lin teaches that the change may be made by performing at least one of changing to a property of the at least one said node, adding one or more additional said nodes as a child to the at least one said node, removing one or more said nodes that are children of the at least one said node, adding an effect to the at least one said node, and removing an effect from the at least one said node, by disclosing that the override video samples may cause a sprite image to become visible and/or disappear and move from its initial disposition to another location while simultanbeously displaying or moving other sprites from a key frame video sample *[column 11, lines 17-26]*.

Claims 26, 29, 31, 32 (System)

4-10. Regarding claim 26, Lin teaches a system comprising a plurality of media and plurality of applications, by disclosing transforming slides for a slide show presentation into movie data and saving the movie data in a movie file for display as a movie by any program capable of playing the movie file *[column 1, lines 6-12]*. Movie data comprises

slide video samples, transition video samples, effect samples, and multiple types of media such as video, sound, and sprite in which references to them are disposed on a video track contained in a video media *[column 3, lines 11-37]*.

Lin teaches an infrastructure layer that provides an API for the plurality of applications which exposes a media timeline that describes one or more presentations of the plurality of media, by disclosing that a movie application programming interface may be selected to save the movie data in the movie file *[column 3, lines 3-10]*.

Lin teaches managing rendering of the one or more presentations, wherein each said presentation describes rendering of said media for a particular interval of time, by disclosing that the video track references a segment on the track occupied by a sample to a corresponding order and timing of the corresponding objects in the slide show presentation. Each track is a timeline with marks (references) that denote when the movie API should begin playing samples from an identified media container *[column 10, lines 43-54]*. Thus, each media container represents a presentation of a particular media.

Lin teaches wherein each said presentation describes a collection of software components configured for dynamic loading such that the collection of software components provide the described rendering of the media for the particular interval of time, by disclosing that each track is a timeline with marks (references) that denote when the movie API should begin playing samples from an identified media container *[column 10, lines 43-54]*. Software components include transitions and effects *[column 14, lines 44-55]*, audio data such as background audio *[column 12, lines 53-65]*, voice

narration *[column 13, lines 3-17]*, and sound effects *[column 13, lines 18-37]*, and video clips from other applications *[column 18, lines 20-24]*. The user may edit the movie data *[column 3, lines 38-44; column 4, lines 13-17; column 9, lines 16-20; figure 2]*.

Lin teaches wherein the collection of software components include a transform by disclosing that the movie API provides editing the movie data such as producing/calling up a video effect including fade, wipe, move, swivel, and spiral as well as selecting the number of colors, size, and position *[column 3, lines 38-50]*. Lin teaches wherein the collection of software components comprise at least one of a timeline source, a media source, a media session, a media engine, a source resolver, and a media sink, by disclosing using a track data structure that references the segment on the track occupied by a sample to the corresponding order and timing of the corresponding object in the presentation. Each track is a timeline with marks that denote when the movie API should begin playing samples from an identified media *[column 10, lines 43-54]*.

4-11. Regarding claim 29, Lin discloses system as described in claim 26, wherein the collection does not change for the particular interval of time described *[column 8, lines 11-20]*.

4-12. Regarding claim 31, Lin discloses the system as described in claim 26, wherein the media timeline includes a plurality of nodes and at least two said nodes reference respective said media, by disclosing movie data comprises slide video samples,

transition video samples, effect samples, and multiple types of media such as video, sound, and sprite in which references to them are disposed on a video track contained in a video media *[column 3, lines 11-37]*. Thus, the references to the movie data represent the plurality of nodes.

Lin teaches at least one said node is configured to reference an effect to be applied to an output of media referenced by the node *[column 10, lines 55-65]*.

4-13. Regarding claim 32, Lin discloses the system as described in claim 26, wherein the media timeline includes a plurality of nodes and at least two said nodes reference respective said media, by disclosing movie data comprises slide video samples, transition video samples, effect samples, and multiple types of media such as video, sound, and sprite in which references to them are disposed on a video track contained in a video media *[column 3, lines 11-37]*.

Lin teaches wherein at least one said node is configured for communication of events to another said node such that a change may be made to the media timeline while the media timeline is rendered *[column 19, lines 46-52]*.

Claims 33, 34, 36-38, 41 (Timeline Source)

4-14. Regarding claim 33, Lin teaches means for dividing a media timeline into one or more presentations each describing a rendering of one or more media during a particular interval of time, by disclosing transforming slides for a slide show presentation into movie data and saving the movie data in a movie file for display as a movie by any

program capable of playing the movie file *[column 1, lines 6-12]*. A video track references a segment on the track occupied by a sample to a corresponding order and timing of the corresponding objects in the slide show presentation. Each track is a timeline with marks (references) that denote when the movie API should begin playing samples from an identified media container *[column 10, lines 43-54]*. Thus, each media container represents a presentation of a particular media.

Lin teaches wherein the media timeline exposes a plurality of nodes to a plurality of applications, wherein one or more said nodes reference respective said media, by disclosing that movie data comprises slide video samples, transition video samples, effect samples, and multiple types of media such as video, sound, and sprite in which references to them are disposed on a video track contained in a video media *[column 3, lines 11-37]*.

Lin teaches wherein the media timeline is configured for dynamic loading such that metadata included in at least one said node specifies a collection of said nodes to be loaded when the media timeline is rendered, by disclosing that each track is a timeline with marks (references) that denote when the movie API should begin playing samples from an identified media container *[column 10, lines 43-54]*. The user may edit the movie data *[column 3, lines 38-44; column 4, lines 13-17; column 9, lines 16-20; figure 2]*.

Lin teaches means for determining a topology for each said presentation, wherein the topology references a collection of software components that, when executed, provides the rendering, by disclosing software components such as

transitions and effects *[column 14, lines 44-55]*, audio data such as background audio *[column 12, lines 53-65]*, voice narration *[column 13, lines 3-17]*, and sound effects *[column 13, lines 18-37]*, and video clips from other applications *[column 18, lines 20-24]* that make up the presentation as a whole.

Lin teaches media processor means for executing the topology for each said presentation that is described by the timeline, by disclosing the playback of movies *[figures 13, 15A,B, 16A,B]*.

4-15. Regarding claim 34, Lin discloses timeline source as described in claim 33, wherein each said collection does not change for the particular interval of time described by a respective said presentation *[column 8, lines 11-20]*.

4-16. Regarding claim 36, Lin discloses method and source as described in claim 33, wherein at least one said node is configured to reference an effect to be applied to an output of said media referenced by the node *[column 10, lines 55-65]*.

4-17. Regarding claim 37, Lin discloses method and source as described in claims 33, wherein at least one said node is configured for communication of events to another said node such that a change may be made to a property of the at least one node while the media timeline is rendered *[column 19, lines 46-52]*.

4-18. Regarding claim 38, Lin discloses timeline source as described in claim 33, wherein the media timeline is configured for dynamic creation such that at least one said node is created while the media timeline is rendered *[column 11, lines 9-16]*.

4-19. Regarding claim 41, Lin discloses timeline source as described in claim 33, further comprising means for translating a time specified by one said node for rendering the one said node with respect to a time specified by another said node *[fig. 2: items 66, 78B, 78C and 98]*.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 4, 9, 15, 23, 30, 35, and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lin et al. (U.S. Patent No. 6,369,835 B1) in view of French et al. (U.S. Patent No. 6,266,053 B1).

Claims 4, 9

Claim 15

Claim 23

6-1. Regarding claims 4, 15, and 23, Lin teaches the invention with respect to claims 1, 12, and 20 respectively. Although Lin teaches, "dividing the media timeline into the one or more presentations" *[column 10, lines 43-48]*, Lin does not expressly teach "each said presentation describes a respective partial topology of software components; and the respective partial topology is for resolving into a full topology that references each software component utilized to provide a respective said presentation." French does teach topology *[column 10, lines 17-24: graph]*. Therefore, it would have been obvious to one having ordinary skill in the art at the time of the invention was made to include partial or full topology in Lin's systems. One would have been motivated to do so in order to efficiently utilized topology on presentation *[column 9, lines 24-30: graph]*.

6-2. Regarding claim 9: Lin teaches the invention with respect to claim 1. Lin does not expressly teach "node is specified as read-only", but French does teach input object is read-only *[column 9, lines 24-30]*. Therefore, it would have been obvious to one having ordinary skill in the art at the time of the invention was made to include read-only in Lin's systems. One would have been motivated to do so in order to efficiently protect node *[column 9, lines 24-30: object]* being override.

Claim 30

6-3. Regarding claim 30: Lin teaches the invention with respect to claim 26. Although Lin discloses "dividing the media timeline into the one or more presentations" *[column 10, lines 43-48]*, Lin does not teach "each said presentation describes a respective

partial topology of software components; and the respective partial topology is for resolving into a full topology that references each software component utilized to provide a respective said presentation." French does teach topology *[column 10, lines 17-24: graph]*. Therefore, it would have been obvious to one having ordinary skill in the art at the time of the invention was made to include partial or full topology in Lin's systems. One would have been motivated to do so in order to efficiently utilized topology on presentation *[column 9, lines 17-24: graph]*.

Claims 35, 40

6-4. Regarding claim 35, Lin teaches the invention with respect to claim 33. Lin does not expressly teach, "dividing the media timeline into the one or more presentations" *[column 10, lines 43-48]* but Lin does not teach "each said presentation describes a respective partial topology of software components; and the respective partial topology is for resolving into a full topology that references each software component utilized to provide a respective said presentation." French does teach topology *[column 10, lines 17-24: graph]*. Therefore, it would have been obvious to one having ordinary skill in the art at the time of the invention was made to include partial or full topology in Lin's systems. One would have been motivated to do so in order to efficiently utilized topology on presentation *[column 9, lines 17-24: graph]*.

6-5. Regarding claim 40: Lin teaches the invention with respect to claim 33. Lin does not expressly teach, "node is specified as read-only", but French does teach input object

is read-only *[column 9, lines 24-30]*. Therefore, it would have been obvious to one having ordinary skill in the art at the time of the invention was made to include read-only in Lin's systems. One would have been motivated to do so in order to efficiently protect node *[column 9, lines 24-30: object]* being override.

Response to Arguments

7. The Examiner acknowledges the Applicant's amendments to claims 1, 12, 20, and 26, and the cancellation of claim 2. Regarding independent claim 1, the Applicant alleges that Lin et al. (U.S. Patent No. 6,369,835 B1), as described in the previous Office action, does not explicitly teach, "wherein the collection of software components include a transform and comprise at least one of a timeline source, a media source, a media session, a media engine, a source resolver, and a media sink, as has been amended to the claims. Contrary to Applicant's arguments, Lin discloses that the movie API provides editing the movie data such as producing/calling up a video effect including fade, wipe, move, swivel, and spiral as well as selecting the number of colors, size, and position *[column 3, lines 38-50]*. Lin also discloses using a track data structure that references the segment on the track occupied by a sample to the corresponding order and timing of the corresponding object in the presentation. Each track is a timeline with marks that denote when the movie API should begin playing samples from an identified media *[column 10, lines 43-54]*. Thus, Lin teaches transforms and a timeline as part of the presentation.

Similar arguments have been presented for claims 12 and 20 and thus, Applicant's arguments are not persuasive for the same reasons.

Regarding independent claim 26, Applicant alleges that Lin, as described in the previous Office action, does not explicitly teach, "wherein the collection of software components include a transform and comprise at least one of a timeline source, a media source, a media session, a media engine, a source resolver, and a media sink and are loaded only when needed, as has been amended to the claims. However, as discussed above, Lin teaches a transform and a timeline source. Additionally, Lin teaches copying objects such as text, graphical images, graphic automation, video, and sound *[column 1, lines 33-36]* of a slide show presentation into the movie file so that an application program that recognizes the objects of the slide show presentation in the movie file may be employed to open the file and produce the presentation *[column 2, lines 60-65]*. Movie data includes a track data structure that references the segment on the track occupied by a sample to the corresponding order and timing of the corresponding object(s) in the slide show presentation. Each track is a timeline with marks (references) that denote when the movie API should begin playing samples **from an identified media (container)** *[column 10, lines 43-54]*. Thus, the objects saved in a particular media will only be loaded and played based on when they are referenced in the timeline. A streaming format is also available for the movie *[column 9, lines 35-36]*.

Similar arguments have been presented for claims 12 and 20 and thus, Applicant's arguments are not persuasive for the same reasons.

Regarding independent claim 33, Applicant alleges that Lin, as described in the previous Office action, does not explicitly teach, "wherein the media timeline is configured for dynamic loading such that metadata included in at least one said node specifies a collection of said nodes to be loaded when the media timeline is rendered". Contrary to Applicant's arguments, Lin discloses using a track data structure that references the segment on the track occupied by a sample to the corresponding order and timing of the corresponding object in the presentation. Each track is a timeline with marks that denote when the movie API should begin playing samples from an identified media *[column 10, lines 43-54]*. Additionally, Lin teaches copying objects such as text, graphical images, graphic automation, video, and sound *[column 1, lines 33-36]* of a slide show presentation into the movie file so that an application program that recognizes the objects of the slide show presentation in the movie file may be employed to open the file and produce the presentation *[column 2, lines 60-65]*. Movie data includes a track data structure that references the segment on the track occupied by a sample to the corresponding order and timing of the corresponding object(s) in the slide show presentation. Each track is a timeline with marks (references) that denote when the movie API should begin playing samples **from an identified media (container)** *[column 10, lines 43-54]*. Thus, the objects saved in a particular media will only be loaded and played based on when they are referenced in the timeline. A streaming format is also available for the movie *[column 9, lines 35-36]*.

Regarding dependent claim 9, Applicant alleges that Lin and French et al (U.S. Patent No. 6,266,053 B1), as described in the previous Office action, do not explicitly

teach that a node is specified as read-only. Contrary to Applicant's arguments, French teaches a similar method of using a timeline with nodes to represent a scene *[column 3, lines 47-64]*. Read-only reference to input objects are used by the operator *[column 9, lines 24-30]*. It would have been obvious to one having ordinary skill in the art at the time of the invention was made to include read-only in Lin's systems. One would have been motivated to do so in order to efficiently protect nodes *[column 9, lines 24-30: object]* from being overridden.

Similar arguments have been presented for claim 40and thus, Applicant's arguments are not persuasive for the same reasons.

Regarding claim 15, Applicant alleges that Lin and French, as described in the previous Office action, do not explicitly teach that each said presentation describes a respective partial topology of software components and the respective partial topology is for resolving into a full topology that references each software component utilized to provide a respective said presentation. Contrary to Applicant's arguments, French does teach topology *[column 10, lines 17-24: graph]*. Therefore, it would have been obvious to one having ordinary skill in the art at the time of the invention was made to include partial or full topology in Lin's systems. One would have been motivated to do so in order to efficiently utilized topology on presentation *[column 9, lines 24-30: graph]*.

Similar arguments have been presented for claims 23 and 30 and thus, Applicant's arguments are not persuasive for the same reasons.

Upon further consideration, Examiner notes that claims 4 and 35 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Lin and French.

Applicant states that dependent claims 4, 6-11, 13-15, 17-19, 23-25, 29-32, 34-38, 40, and 41 recite all the limitations of the independent claims, and thus, are allowable in view of the remarks set forth regarding independent claims 1, 12, 20, 26, and 33. However, as discussed above, Lin is considered to teach claims 1, 12, 20, 26, and 33, and consequently, claims 4, 6-11, 13-15, 17-19, 23-25, 29-32, 34-38, 40, and 41 are rejected.

Conclusion

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to ALVIN H. TAN whose telephone number is (571)272-8595. The examiner can normally be reached on Mon-Fri 10:00-6:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Dennis Chow can be reached on 571-272-7767. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

AHT

Assistant Examiner
Art Unit 2173

/Tadesse Hailu/
Primary Examiner, Art Unit 2173